Langara College

Final Exam

CPSC 2221

Max Marks – 50

Due – 3.45 PM

Name: Jay Seung Yeon Lee ID: 100 357 736

Instructions:

- **Seats will be assigned when you arrive. If you arrive late no extra time will be provided.**
- **Very Important: A working Laptop with PostGRESQL installed and ready for the exam is expected. You are responsible for the system not working.**
- **Remember, if you miss your laptop or if it's not working you are responsible for the grades deducted due to that, Faculty is not responsible for your carelessness to maintain your own system.**
- **No personal notes allowed.**
- **D2L slides/lab solutions/midterm solutions/quiz solutions are allowed to be used during the exam.**
- **Also, postgresql official documentation website can be referred to during the exam.**
- **No other internet browser is allowed, if students are seen browsing or chatting, then the exam will be forfeited with a 0 grade provided for the Final.**

Problem 1. Develop and E-R Diagram based on the following information about the client's small business documents.

Heather Sweeney is an interior designer who specializes in home kitchen design. She offers variety of seminars at home shows, kitchen and appliance stores, and other public locations. The seminars are free; she offers them as a way of building her customer base. She earns revenue by selling books and videos that instruct people on kitchen design. She also offers custom-design consulting services.

After someone attends a seminar, Heather wants to leave no stone unturned in attempting to sell that person one of her products or services. She would therefore like to develop a database to keep track of customers, the seminars they have attended, the contacts she has made with them, and the purchases they have made. She wants to use this database to continue to contact her customers and offer them products and services.

Figure below shows the seminar customer list form that Heather or her assistant fills out at seminars. This includes basic data about seminar as well as name, phone, and email address of each seminar attendee. Each of her seminar has at least more than 10 customers. This gives two Entities, one SEMINAR and the other CUSTOMER.

Heather Sweeney Designs
Seminar Customer List

Date: October 12, 2018 Location: San Antonio Convention Center

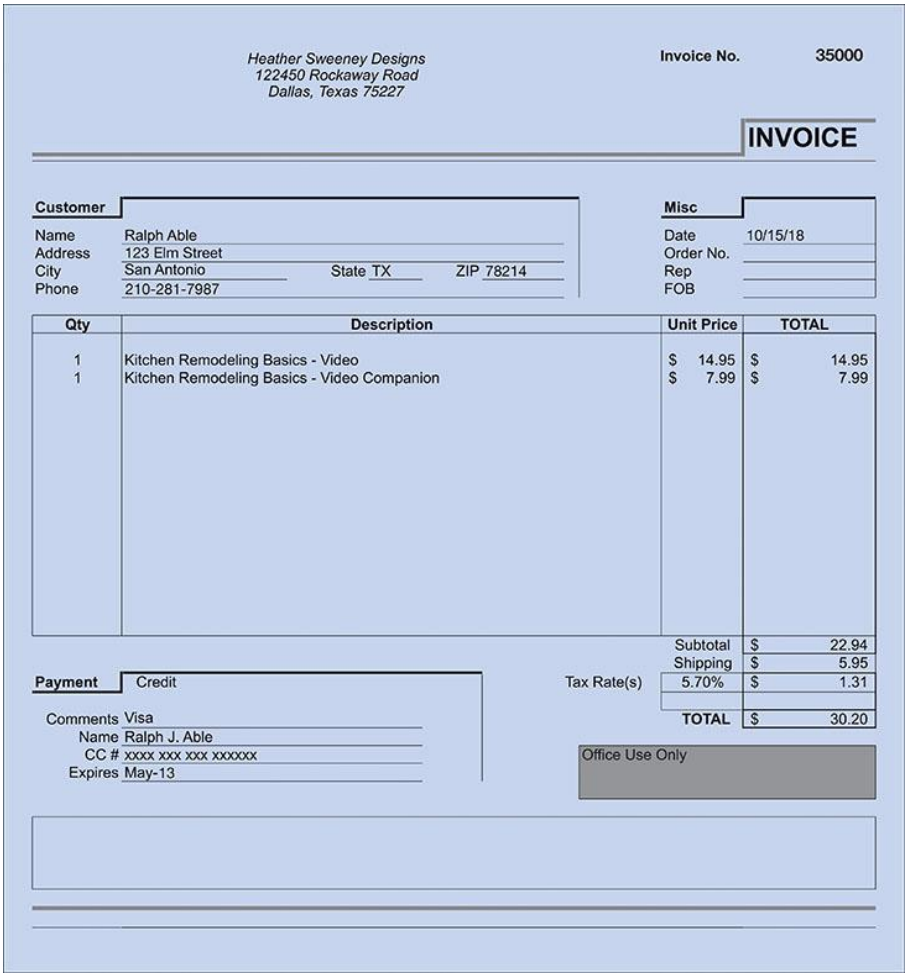
Time: 11 AM Title: Kitchen on a Budget

Name	Phone	Email Address
Nancy Jacobs	817–871–8123	Nancy.Jacobs@somewhere.com
Chantel Jacobs	817–871–8234	Chantel.Jacobs@somewhere.com
Ralph Able	210–281–7987	Ralph.Able@somewhere.com
Etc.		
27 names in all		

The sales invoice that Heather uses to sell books and videos is shown in Figure below. SALES INVOICE itself needs to be an entity, and because the sales invoice has customer data it has a

relationship back to CUSTOMER. (Note that we do not duplicate the customer data because we can obtain data items via the relationship; if data items are missing, we add them to CUSTOMER).

Because Heather runs her computer with minimal security, she decided that she did not want to record credit card numbers in her computer. Instead, she records only the PaymentType value in the database and does not record the rest. Product details are stored in a separate entity PRODUCT with a productid and is related to the INVOICE.



Heather Sweeney Designs
122450 Rockaway Road
Dallas, Texas 75227

Invoice No. 35000

INVOICE

Customer

Name	Ralph Able		
Address	123 Elm Street		
City	San Antonio	State TX	ZIP 78214
Phone	210-281-7987		

Misc

Date	10/15/18
Order No.	
Rep	
FOB	

Qty	Description	Unit Price	TOTAL
1	Kitchen Remodeling Basics - Video	$ 14.95	$ 14.95
1	Kitchen Remodeling Basics - Video Companion	$ 7.99	$ 7.99

Subtotal		$ 22.94
Shipping		$ 5.95
Tax Rate(s)	5.70%	$ 1.31
TOTAL		$ 30.20

Payment Credit

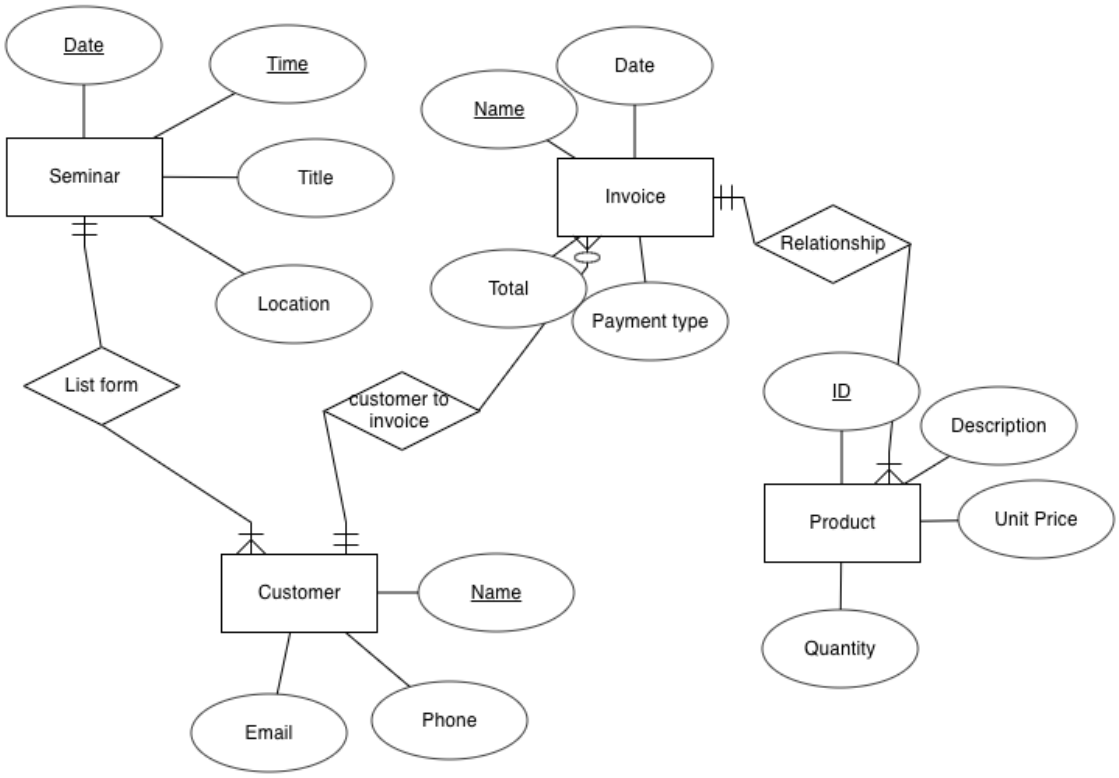
Comments	Visa
Name	Ralph J. Able
CC #	xxxx xxx xxx xxxxxx
Expires	May-13

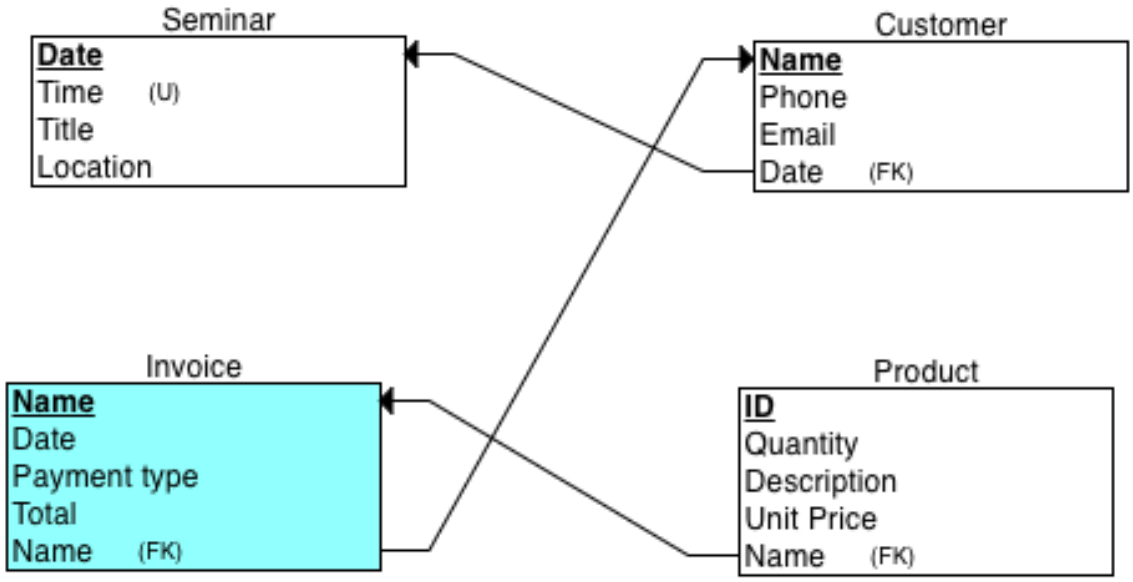
Office Use Only

Based on the information above, please help Heather to construct an **ER Diagram using the ERD plus.**

Both the conceptual schema and the relational schema needs to be pasted here. Do not be biased and only use information provided in the client description. Do not use extra columns unless specified by the client. **[10 Marks]**

Paste your schemas below





Problem 2. Write the following queries in SQL, using the university schema database. The create table queries and the insert data queries have been provided to you for quick start in D2L along with the exam.

a. Find the titles of courses in the Comp. Sci. department that have 3 credits. **[3 Marks]**

Write/paste the query in text format and paste the screenshot of the output below the query

```
SELECT Title
FROM Course
WHERE Dept_Name = 'Comp. Sci.'
AND Credits = 3
```

	title character varying (50)
1	Robotics
2	Image Processing
3	Database System Conce...

b. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result. **[4 Marks]**

Write/paste the query in text format and paste the screenshot of the output below the query

```
SELECT s.id
FROM takes AS s
INNER JOIN Teaches AS t
ON s.course_id = t.course_id
AND s.semester = t.semester
AND s.year = t.year
WHERE t.id = '22222'
GROUP BY s.id
```

	id character varying (5)
1	44553

c. Find the enrollment of each section that was offered in Fall 2017. **[4 Marks]**

Write/paste the query in text format and paste the screenshot of the output below the query

```
SELECT COUNT(id), course_id, sec_id
FROM Takes
GROUP BY course_id, sec_id
```

	count bigint	course_id character varying (8)	sec_id character varying (8)
1	1	HIS-351	1
2	7	CS-101	1
3	2	CS-190	2
4	1	BIO-301	1
5	2	CS-315	1
6	1	BIO-101	1
7	1	PHY-101	1
8	1	CS-319	2
9	1	CS-319	1
10	1	FIN-201	1

d. Find the maximum enrollment, across all sections, in Fall 2017. **[4 Marks]**

Write/paste the query in text format and paste the screenshot of the output below the query

```
SELECT COUNT(id), course_id, sec_id
FROM Takes
GROUP BY course_id, sec_id
ORDER BY COUNT(id)
DESC LIMIT 1
```

	count bigint	course_id character varying (8)	sec_id character varying (8)
1	7	CS-101	1

Problem 3. We will be using 2 tables from the DVD Rental database, the customer table and the rental table. Although it is assumed that the DVD rental database has been restored in your postgresql, it is provided with your exam in case you missed it in the first class.

customer(customer_id, first_name, last_name, address_id) – customer_id (PK)

rental(rental_id, rental_date, customer_id, inventory_id) – rental_id (PK)

We will create a reservation table in the dvdrental Database. This table will be used to reserve DVD'S for customers. Code provided below.

```
CREATE TABLE reservation
(customer_id int,
 inventoy_id int,
 reserve_date date
 )
```

We will alter the reservation table and add primary key and foreign keys. Write the Alter Table to add (customer_id, inventory_id) as the composite primary key for the reservation table.

Also, use 2 more Alter Tables to add the respective Foreign Keys of customer_id and inventory_id.

Write the 3 Alter tables Below **[3 Marks]**
ALTER TABLE Reservation ADD PRIMARY KEY (Customer_id, Inventory_Id); ALTER TABLE Reservation ADD FOREIGN KEY(Customer_Id) REFERENCES Customer(Customer_Id); ALTER TABLE Reservation ADD FOREIGN KEY(Inventory_Id) REFERENCES Rental(Inventory_Id);

We would like to create a trigger that will help to **keep track of all operations** performed on the reservation table. We want to record whether an **insert, or update** occurred on the **reservation** table. The first step that We will do is to create a table to record the operations, I will call the table **reservation_audit**. We will store the first letter of the operation (I for Insert, and U for Update), the timestamp of the operation, and the values of the operation.

Here is the definition of the **reservation_audit**:

```
CREATE TABLE reservation_audit(
    operation         char(1)   NOT NULL,
    stamp             timestamp NOT NULL,
    customer_id            text      NOT NULL,
    inventory_id           text      NOT NULL,
    res_date               date      NOT NULL
);
```

We will create a **reservation_audit** trigger on the **reservation table** (ON reservation) that will be triggered **after Insert, Update or Delete** (AFTER INSERT OR UPDATE). The trigger will call the **process_reservation_audit** function. The **process_reservation_audit()** will be responsible to record in the **reservation_audit** table the transaction performed whether it is an **Insert, or Update.**

Write the query for the reservation_audit trigger below **[3 Marks]**

```
CREATE TRIGGER reservation_audit
AFTER INSERT OR UPDATE
ON Reservation
FOR EACH ROW
EXECUTE PROCEDURE process_reservation_audit();
```

Write the query for the process_reservation_audit() function below **[4 Marks]**

[**Hint:** From postgresql documentation. TG_OP

> Data type text; a string of INSERT, UPDATE, DELETE, or TRUNCATE telling for which operation the trigger was fired.]
>
> TG_OP = 'INSERT' means it will check if an INSERT command is called. TG_OP is a psql keyword and not a custom variable. So, use it as such. Refer to the link below for more examples similar to what you need to solve this question:

https://www.postgresql.org/docs/current/plpgsql-trigger.html]

```
CREATE OR REPLACE FUNCTION process_reservation_audit()
RETURNS trigger AS
$$
BEGIN
IF NEW.operation <> OLD.operation THEN
INSERT INTO reservation_audit(operation,customer_id,inventory_id, stamp, res_data)
VALUES(NEW operation, OLD.customer.id, OLD.inventory_id, OLD stamp, OLD res_data);
END IF
RETURN NEW;
END;
$$
LANGUAGE PLPGSQL;
```

We will insert a record in the **reservation** table:

```
INSERT INTO reservation (customer_id, inventoy_id,reserve_date) values (2,13, CURRENT_DATE);
```

We will select from the reservation and the reservation_audit tables. We will see a new record in the reservation table. We will also see that a new record is added in the reservation_audit table indicating that an insert (I) operation has occurred in the reservation table.

```
SELECT * FROM reservation;

SELECT * FROM reservation_audit;
```

Show the screenshots of running the 2 queries below **[3 Marks]**

We will update a record in the **reservation** table as follows.

```
UPDATE reservation
SET inventoy_id = 18
WHERE customer_id = 2 AND inventoy_id =15;

SELECT * FROM reservation_audit;
```

A row is updated in the reservation table. A record is added in the reservation_audit table indicating that an update (U) operation has occurred on the reservation table.

Show the screenshots of running the select * from reservation_audit below **[2 Marks]**

Problem 4. Database transactions are governed by the ACID properties (Atomicity, Consistency, Isolation & Durability). Consider the following Transactions applied on a Database table TestDB? Suppose there are 2 rows X and Y. Also assume one of the rows of X has value 500 and one of the rows of Y has value 100 (The values are provided just for reference). Consider two transactions **T** and **T"** performed on X and Y.

T	T"
Read (X) X: = X*100 Write (X) Read (Y) Y: = Y – 50 Write (Y)	Read (X) Read (Y) Z: = X + Y Write (Z)

Suppose transaction T has been executed till Read (Y) when T'' starts. This results in an error. From the ACID properties, find out which property this operation on T and T'' violates and what needs to be done to correct it. The example values of X and Y as mentioned above can be used to solve this question. **[5 Marks]**

Write Your answer in the space provided.
This would be an example of "Temporary update problem", and it is a violation of Isolation as the problem is occurring from transactions happening concurrently and by the time and the value of Y is updated as T'' is reading Y, causing value of Y to be incorrect. Easiest solution for this problem would be to run T and T'' asynchronously, but if the procedures were to be ran concurrently then Isolation property should be strictly enforced to avoid this problem. When Isolation property is strictly enforced, it will hide it's updates from the other transactions until it is fully committed, therefore T'' will not have issue with reading Y.

Problem 5. Let V be a view created over relation R (create view V as SELECT . . .FROM.. .). Assume that initially Bob has all superuser permissions on R (including permission to grant permissions to others), nobody else has GRANT permissions on R, and that Alice and Clara have select permission on V. Now consider the sequence of commands executed by the specified users to grant and revoke permissions as showed in Table below: **[5 Marks]**

Order	Command	Executed by
1	Grant Select on *R* To Alice with Grant Option	Bob
2	Grant Select on *R* To Clara	Alice
3	Grant Select on *R* To Donald	Alice
4	Grant Select on *R* To Clara	Bob
5	Revoke Select on *R* From Alice	Bob

Question: Which of *Bob*, *Alice*, *Clara*, *Donald* are authorized to execute each of the commands as showed *below* Example: If BOB can execute commands 1 and 3, then write BOB '-' (hyphen) has access to (1,3). Similarly, you will write authorization access for all 4 of the staff users.

1. SELECT X FROM R WHERE $Y < 100$
2. UPDATE R SET $Y = Y * 3$
3. SELECT A FROM V WHERE $C = 10$
4. CREATE VIEW $View2$ AS SELECT * FROM R

Answer Here:

Bob - (1,2,3,4)

Alice - (none)

Clara - (1, 3)

Donald - (1,3)